3. Sum Attack

Consider each unique IP address as representing a different attacker. How many distinct attackers gained access to the system

```
┌──(imen㉿hbtn-lab)-[…/web_application_security/0x0c_web_application_foresics]
└─$ ./3-ips.sh
18
```

Command Breakdown:

```
grep "Accepted password for root" auth.log | grep -Eo '[0-9]{1,3}(\.[0-9]{1,3}){3}' | sort -u | wc -l
```

1. **`grep "Accepted password for root" auth.log`**:
   - This filters the `auth.log` file to find lines that contain the string **"Accepted password for root"**. This string indicates that the `root` user successfully authenticated using a password. It will return all such log entries that show successful login attempts for the `root` user.
2. **`grep -Eo '[0-9]{1,3}(\.[0-9]{1,3}){3}'`**:
   - This command uses `grep` with the **`-E`** option (for extended regular expressions) and the **`-o`** option (to only output the matched part of the line).
   - The regular expression `'[0-9]{1,3}(\.[0-9]{1,3}){3}'` matches **IPv4 addresses** in the form of `xxx.xxx.xxx.xxx`, where `xxx` is a number between 0 and 255. This will extract all the IP addresses from the filtered `auth.log` entries.
     - `[0-9]{1,3}` matches a sequence of 1 to 3 digits.
     - `(\.[0-9]{1,3}){3}` matches three additional segments, each starting with a dot `.` followed by 1 to 3 digits.
3. **`sort -u`**:
   - This sorts the extracted IP addresses in **ascending order** and removes any **duplicate IP addresses** (`-u` stands for unique). This will leave you with a list of distinct IP addresses that have attempted to authenticate as the `root` user.
4. **`wc -l`**:
   - This command counts the number of lines in the input, which in this case corresponds to the number of **unique IP addresses** that have successfully logged in as `root`. Essentially, it gives the total count of distinct IPs that have performed a successful login for `root`.

What the command does:

This entire command sequence counts how many unique IP addresses have attempted to authenticate as the `root` user using a password, based on the entries in the `auth.log` file.

Example scenario:

If your `auth.log` contains entries like:

```
Feb 24 14:23:56 server sshd[2345]: Accepted password for root from 192.168.1.10 port 22
Feb 24 14:23:59 server sshd[2345]: Accepted password for root from 192.168.1.11 port 22
Feb 24 14:24:05 server sshd[2345]: Accepted password for root from 192.168.1.10 port 22
Feb 24 14:24:10 server sshd[2345]: Accepted password for root from 10.0.0.5 port 22
```

This command will:

1. Filter for lines with `"Accepted password for root"`.
2. Extract the IP addresses: `192.168.1.10`, `192.168.1.11`, and `10.0.0.5`.
3. Remove duplicates and sort: `192.168.1.10`, `192.168.1.11`, and `10.0.0.5`.
4. Count the number of unique IPs: `3`.

The final output will be:

```
3
```

Use case:

This command is useful to identify how many unique IP addresses have successfully authenticated as the `root` user. It can help in detecting unauthorized or suspicious logins, especially if unexpected IP addresses are observed.